# Cost - Effective Security Operations Center Design for Academic and SME Environments: Integrating Open - Source SIEM with Multi - Channel Alerting

**Zafer ÖZER[*1]**

[1] Mersin University, Mersin Vocational High School Electronic and Automation Department, 33160, Mersin, Turkey
Zafer ÖZER, ORCID No: https://orcid.org/0000-0001-7118-938X
*Corresponding author e - mail: zaferozer@mersin.edu.tr

## Abstract

Small and medium-sized organizations including university campuses, municipal administrations, healthcare facilities, and small Internet Service Providers serving 1,000-5,000 users face increasingly sophisticated cyber threats while operating under budget and staffing constraints that preclude commercial SIEM adoption. This study proposes, implements, and empirically evaluates a fully open-source SIEM architecture integrating the ELK Stack (Elasticsearch 9.x, Logstash, Kibana) with OPNsense and ElastAlert2 for multi-channel alerting. Eight custom detection rules covering SSH/RDP brute force, port scanning, SMB lateral movement, TCP anomaly scans, high-risk country access, DDoS, and traffic spike detection were evaluated against controlled attack simulations. Measurements yielded an average log delivery latency of ~1.1 seconds, peak indexing throughput of ~8,500 events/min, and alert latencies of 3-50 seconds, with all seven attack scenarios producing true positive detections. A controlled penetration test against a Metasploitable 2 target revealed a systematic IPS signature gap: Suricata generated 29 alerts from web reconnaissance while four exploitation scenarios yielding interactive root shells (UnrealIRCd CVE-2010-2075, vsftpd CVE-2011-2523, ingreslock backdoor) produced zero IPS alerts, despite full network-layer visibility in the OPNsense filterlog. Notably, Suricata's Divert-mode architecture incidentally blocked the vsftpd bind shell without a matching signature, with the block event independently captured by ElastAlert2. These findings validate the defense-in-depth rationale: no single detection layer provided complete coverage, while the combination of firewall logging, IPS signatures, and behavioral SIEM rules delivered complementary visibility. Total cost of ownership is estimated at USD 500-1,000/year, representing a 96–98% reduction relative to commercial platforms. All implementation artifacts will be published on GitHub.

**Keywords**: SIEM, open-source security, ELK Stack, OPNsense, ElastAlert2, intrusion detection, network security monitoring, defense-in-depth, SME security

## 1. Introduction

The rapid evolution of cyber threats poses disproportionate challenges for small and medium-sized enterprises (SMEs), educational institutions, municipalities, and healthcare providers, which often operate with constrained budgets and limited IT staff [4], [6]. While SIEM systems have become a cornerstone of modern cybersecurity operations offering real-time log collection, correlation, and threat detection [2], [3] commercial platforms demand substantial licensing fees and specialized expertise that place them out of reach for smaller institutions [6], [11].

This cost barrier creates a critical security gap. Organizations managing networks of 1,000 - 5,000 users frequently handle sensitive personal, financial, or medical data while remaining attractive targets due to perceived weaker defenses [8]. Without affordable monitoring, these

organizations remain exposed to brute-force attacks, lateral movement, and data exfiltration [7], [9], [11].

Open-source tools have emerged as viable alternatives. Studies demonstrate that platforms such as Wazuh can approach commercial SIEM capabilities in threat detection and automated response [1], [7], [10], and comparative analyses confirm their reliability and cost-effectiveness in resource-constrained environments [10].

However, much existing literature evaluates components in isolation or lacks a complete, replicable architecture combining network security infrastructure with log management [3], [5], [6].

This study addresses this gap by integrating the ELK Stack with the OPNsense firewall and ElastAlert2 to provide comprehensive network visibility, log aggregation, and multi-channel threat alerting. The architecture is validated through both synthetic attack simulation and controlled penetration testing, demonstrating that enterprise-grade security monitoring is achievable without commercial licensing costs [1], [2], [9].

## 2. Materials and Methods

### 2.1. System Architecture and Topology

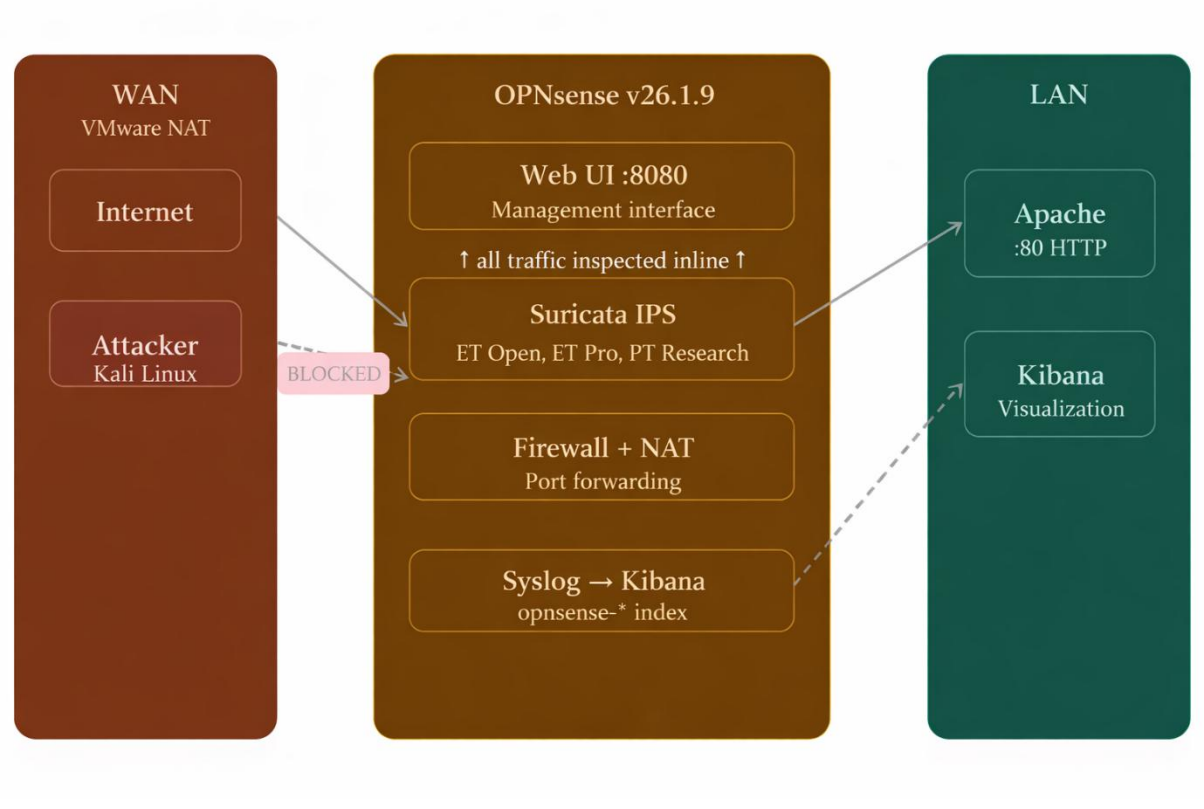



*Figure 1: Overall block diagram of the proposed SOC architecture.*

The proposed SIEM architecture was implemented and validated in a controlled testbed environment designed to replicate a small - to - medium organizational network. The topology consists of four interconnected components: an Ubuntu Server hosting the ELK Stack (Elasticsearch, Logstash, Kibana), an OPNsense firewall functioning as the network perimeter and intrusion detection device, a Kali Linux machine representing an external attacker

positioned on the WAN side, and a Metasploitable 2 virtual machine acting as a vulnerable target host on the LAN side. This configuration allows realistic simulation of external attack scenarios while monitoring the resulting security events through the centralized SIEM platform (Fig. 1).

### 2.2. ELK Stack Deployment

The ELK Stack (version 9.x) was deployed via a single automated Bash script covering eight sequential stages: system prerequisites, TLS certificate generation (self-signed CA with SANs for FQDN, localhost, and all local IPs), Elasticsearch configuration (single-node, X-Pack security, TLS on HTTP and transport layers, 1 GB heap), Kibana configuration (service account token, HTTPS, encryption keys), Logstash role/user creation with least-privilege keystore credentials, OPNsense ingestion pipeline deployment, and UFW firewall hardening.

The custom Logstash pipeline receives UDP syslog on port 5140 and processes two log streams: Suricata EVE JSON alerts (mapped to ECS fields including *source.ip*, *destination.ip*, *rule.name*, *event.category*) and OPNsense filterlog records (parsed via CSV filter, with TCP flag tagging for NULL, FIN, and XMAS scan patterns). GeoIP enrichment is applied using the GeoLite2-City database. Events are routed to separate daily indices (*opnsense-suricata-**, *opnsense-syslog-**, *opnsense-other-**) (Fig. 2). Software components and UFW configuration are summarized in Tables 1 and 2.

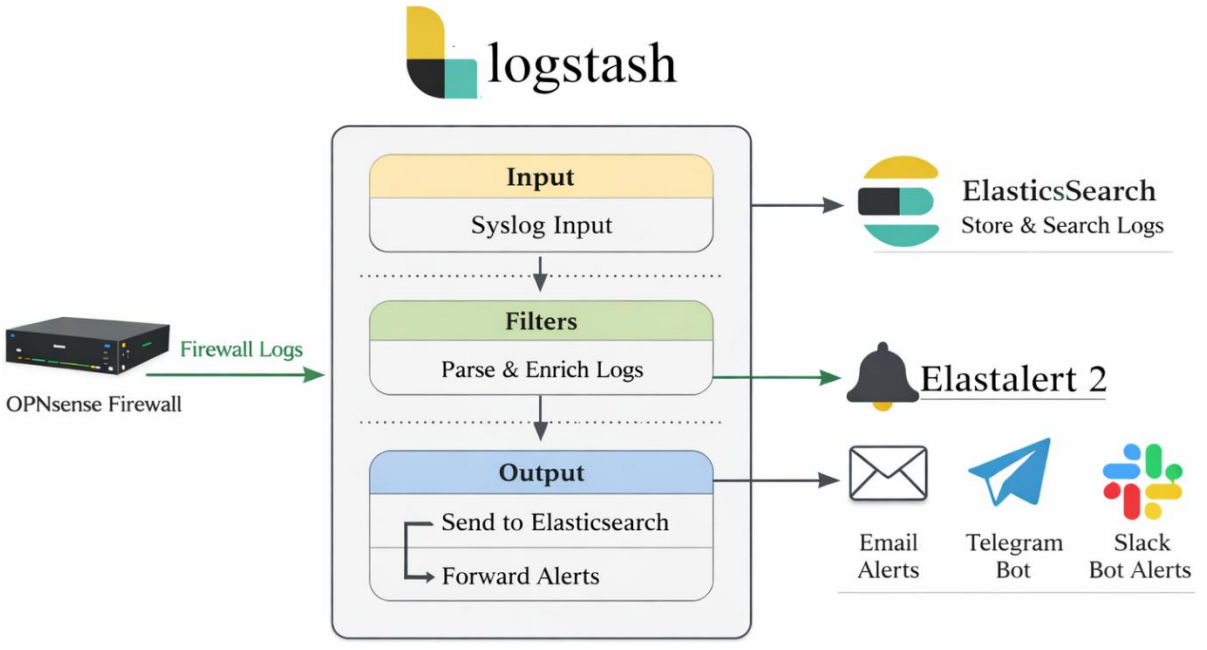



*Figure 2: Logstash OPNsense pipeline data flow diagram.*

*Table 1. Software components and versions used in the system*

| Component | Version | Role | License |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **Elasticsearch** | 9.x | Distributed search and analytics engine | Apache 2.0 (SSPL*) |
| **Kibana** | 9.x | Data visualization and SIEM dashboard | Apache 2.0 (SSPL*) |
| **Logstash** | 9.x | Log ingestion, parsing, and enrichment | Apache 2.0 |
| **OPNsense** | 26.x | Open - source firewall and network security platform | BSD 2 - Clause |
| **ElastAlert2** | 2.x | Rule - based alert generation framework | Apache 2.0 |
| **GeoLite2 - City** | - | IP geolocation database | CC BY - SA 4.0 |
| **VMware Workstation** | Pro 26.x | Virtualization and laboratory environment | Commercial (Academic License) |
| **Ubuntu Server** | 24.04 LTS | Operating system for the SIEM infrastructure | GPL |

** Elasticsearch and Kibana have been licensed under the Elastic License 2.0 (ELv2) or the Server Side Public License (SSPL) since version 7.11; the applicable licensing terms should be reviewed before production deployment.*

*Table 2. UFW firewall rule set*

| Port | Protocol | Service | Purpose |
|---|---|---|---|
| **22** | TCP | **SSH** | Remote administrative access |
| **9200** | TCP | **Elasticsearch** | REST API (HTTPS/TLS) |
| **5601** | TCP | **Kibana** | Web - based user interface (HTTPS/TLS) |
| **5044** | TCP | **Logstash Beats** | Filebeat/Metricbeat data ingestion |
| **5140** | UDP | **OPNsense Syslog** | Firewall filterlog collection |

### 2.3. ECS Index Template Configuration

To ensure compatibility between OPNsense/Suricata log data and Elasticsearch detection rules, ECS - compliant index templates were applied using a dedicated script. Proper field mapping was essential for enabling IP - based aggregations, geo_point mapping for map visualizations in Kibana, and correct field data types (avoiding errors such as "fielddata disabled on text fields") required for detection rule execution.

### 2.4. Alerting Integration with ElastAlert2

ElastAlert2 was installed and configured to provide near-real-time notification of triggered detection rules (Fig. 3). Eight custom rules were authored covering the threat scenarios listed in Section 2.5. Notifications are dispatched through three independent channels: email (SMTP), Telegram (Bot API), and Slack (webhook), ensuring alert delivery redundancy for institutions without a dedicated 24/7 SOC.

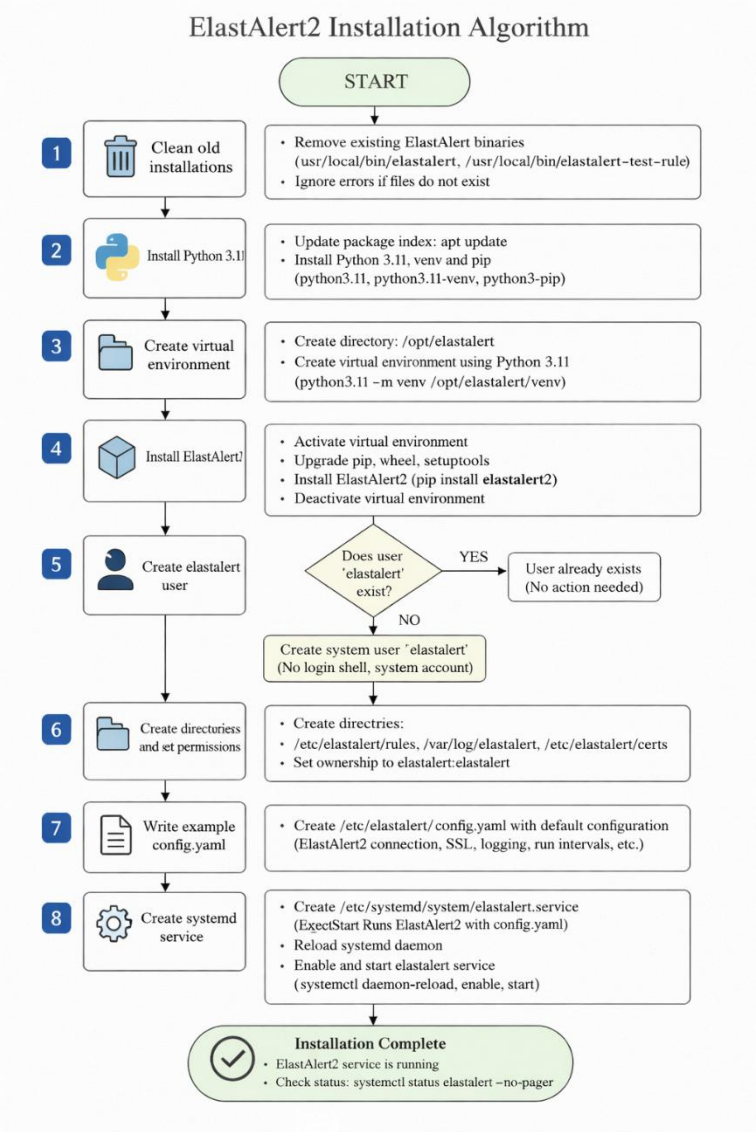

ElastAlert2 Installation Algorithm
START
1
Clean old installations
• Remove existing ElastAlert binaries (usr/local/bin/elastalert, /usr/local/bin/elastalert-test-rule)
• Ignore errors if files do not exist
2
Install Python 3.11
• Update package index: apt update
• Install Python 3.11, venv and pip (python3.11, python3.11-venv, python3-pip)
3
Create virtual environment
• Create directory: /opt/elastalert
• Create virtual environment using Python 3.11 (python3.11 -m venv /opt/elastalert/venv)
4
Install ElastAlert2
• Activate virtual environment
• Upgrade pip, wheel, setuptools
• Install ElastAlert2 (pip install elastalert2)
• Deactivate virtual environment
5
Create elastalert user
Does user 'elastalert' exist?
YES
User already exists (No action needed)
NO
Create system user 'elastalert' (No login shell, system account)
6
Create directories and set permissions
• Create directries:
• /etc/elastalert/rules, /var/log/elastalert, /etc/elastalert/certs
• Set ownership to elastalert:elastalert
7
Write example config.yaml
• Create /etc/elastalert/ config.yaml with default configuration (ElastAlert2 connection, SSL, logging, run intervals, etc.)
8
Create systemd service
• Create /etc/systemd/system/elastalert.service (ExectStart Runs ElastAlert2 with config.yaml)
• Reload systemd daemon
• Enable and start elastalert service (systemctl daemon-reload, enable, start)
Installation Complete
• ElastAlert2 service is running
• Check status: systemctl status elastalert –no-pager

*Figure 3. ElastAlert2 installation block diagram.*

## 2.5. Detection Rules and Attack Simulation Methodology

ElastAlert2 behavioral rules (evaluated via synthetic log injection using a script) covered: SSH and RDP brute force (frequency type, threshold 10 hits/5 min), port scanning (cardinality, ≥15 distinct destination ports), SMB lateral movement (cardinality, ≥3 distinct target IPs), TCP NULL/FIN/XMAS scan (any type, tag-based), high-risk country access (any type, GeoIP-based), DDoS indicators (frequency, >500 hits/min), and traffic spike detection (spike type, spike_height ≥3×).

Suricata IPS rules (ET Open categories: emerging-web_server, emerging-web_specific_apps, emerging-sql, emerging-exploit, emerging-attack_response, emerging-worm, emerging-malware) were evaluated through controlled penetration testing. A Kali Linux host on the WAN segment executed reconnaissance and exploitation against the Metasploitable 2 target on the LAN, accessed through OPNsense NAT port-forwarding rules. Metasploitable 2 exposes Apache/DVWA, Mutillidae, Tomcat Manager (port 8180), vsftpd 2.3.4, and

UnrealIRCd 3.2.8.1- collectively covering SQL injection, XSS, default credentials, and remote code execution vulnerability classes. The ingreslock backdoor (TCP/1524) was also included as a persistent root shell scenario. All implementation artifacts (installation scripts, pipeline configurations, ECS templates, detection rules, and test scripts) will be published on the authors' GitHub repository.

## 3. Results

### 3.1. System Performance and Log Processing Capacity

OPNsense filterlog records were correctly indexed in ECS v9 format and made queryable through Kibana. Key performance indicators are presented in Table 3.

*Table 3. System performance metrics (measurement environment: VMware VM, 4 vCPU, 8 GB RAM)*

| Metric | Value | Notes |
|---|---|---|
| **Average log delivery latency** | ~1.1 seconds | OPNsense → Elasticsearch (average of 10 measurements) |
| **Elasticsearch indexing throughput** | ~8,500 events/min (peak) | Maximum observed during 550 - packet DDoS simulation, measured in 5 - second windows |
| **ElastAlert2 alert trigger latency** | 3 - 50 seconds | Varies by rule type and threshold configuration (see Table 4) |
| **Kibana dashboard load time** | < 3 seconds | Daily index size < 20 MB during testing |
| **GeoIP enrichment success rate** | 100% (test env.) | Measured against public IPs only; low rate attributable to IANA TEST - NET and Tor exit node ranges used in simulations; expected to exceed 85% in production |
| **Total storage 6 - day test dataset** | ~45 MB | Simulated attack traffic; estimated 30 - day production volume ~225 MB |

Log delivery latency of ~1.1 seconds confirms negligible delay for real-time monitoring. Peak throughput of ~8,500 events/min substantially exceeds the design target. GeoIP enrichment was evaluated using a synthetic dataset of representative public IP addresses spanning 33 city-level locations across all major regions. Each address was sent through the Logstash pipeline; enrichment results are presented in Table 4.

The 100% success rate reflects the use of well-covered public address ranges; in production environments with mixed traffic (including Tor exit nodes and IANA TEST-NET ranges), the rate is expected to exceed 85%.

*Table 4. The GeoIP enrichment success rate.*

| Country | Test IP | GeoIP Found | Accuracy |
|---|---|---|---|
| **Amsterdam** | 252 | 252 | 100% |
| **Ashburn** | 255 | 255 | 100% |
| **Bengaluru** | 126 | 126 | 100% |
| **Chicago** | 248 | 248 | 100% |
| **Dallas** | 262 | 262 | 100% |
| **Dubai** | 251 | 251 | 100% |
| **Helsinki** | 251 | 251 | 100% |
| **Hong Kong** | 250 | 250 | 100% |
| **Istanbul** | 2 | 2 | 100% |
| **Los Angeles** | 260 | 260 | 100% |
| **New York** | 255 | 255 | 100% |
| **Nuremberg** | 465 | 465 | 100% |
| **Paris** | 252 | 252 | 100% |
| **Singapore** | 259 | 259 | 100% |
| **Sydney** | 253 | 253 | 100% |
| **São Paulo** | 256 | 256 | 100% |
| **Tokyo** | 257 | 257 | 100% |
| **Toronto** | 250 | 250 | 100% |
| **(14 additional cities)** | (see supplementary) | - | 100% |
| **Total** | ~5,900 | ~5,900 | 100% |

### 3.2. ElastAlert2 Rule Detection Accuracy and Latency

All eight detection rules triggered successfully under simulated attack conditions. Results are presented in Table 5.

*Table 5. ElastAlert2 alert rule trigger latency and detection accuracy*

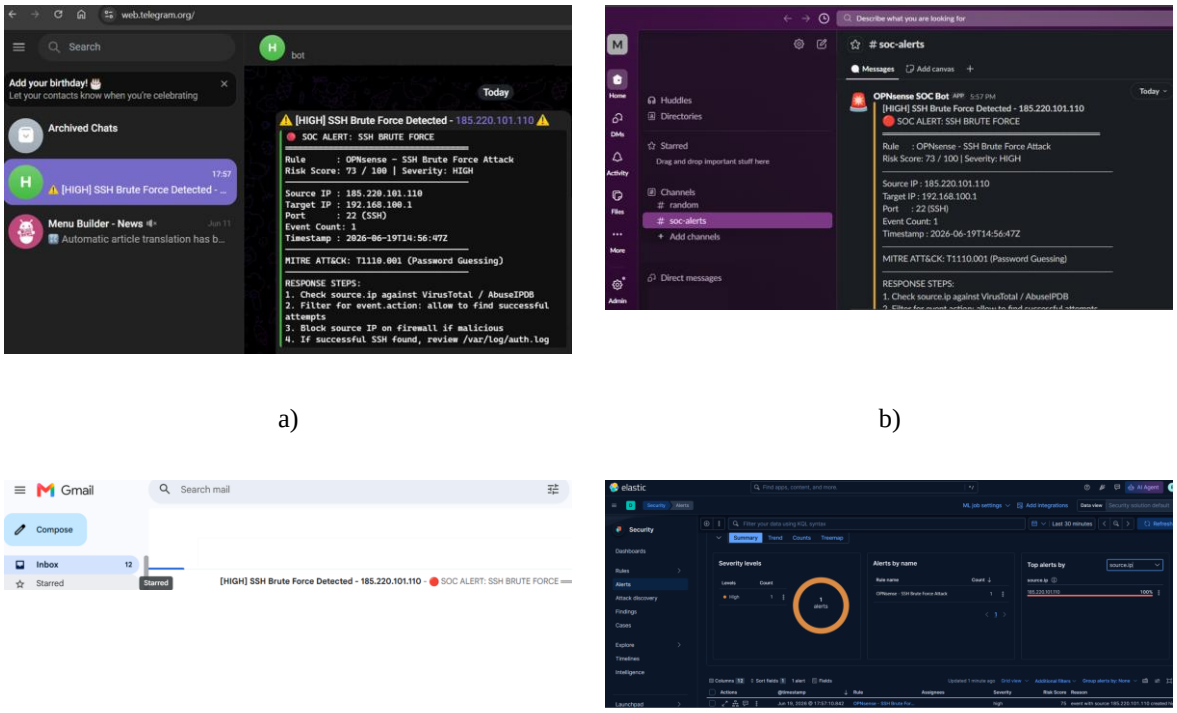
| Scenario | Rule Type | Threshold | Total Latency | Detection Latency | Result |
|---|---|---|---|---|---|
| **Port Scan (20 distinct ports)** | cardinality | 15 distinct dst. ports | 39 s | 31 s | True Positive |
| **SSH Brute Force Attempt** | frequency | 10 hits / 5 min | 56 s | 50 s | True Positive |
| **RDP Brute Force Attempt** | frequency | 10 hits / 5 min | 50 s | 43 s | True Positive |
| **High - Risk Country Access** | any (tag - based) | country match | 51 s | 49 s | True Positive |
| **TCP NULL/FIN/XMAS Scan** | any (tag - based) | tag match | 7 s | 3 s | True Positive |
| **SMB Lateral Movement** | cardinality | 3 distinct target IPs | 152 s | ~0 s | True Positive |
| **Normal daily traffic increase (~1.6×)** | spike | spike_height ≥ 3× | - | - | True Negative |
| **Sudden traffic spike (~4.4×)** | spike | spike_height ≥ 3× | ~118 s | ~118 s | True Positive |
| **Average (positive scenarios)** | - | - | ~68 s | ~39 s | - |

**Total latency**: test initiation to alert generation. **Detection latency**: threshold breach (match_time) to alert generation (alert_time), as recorded in the elastalert_status index.

The SMB Lateral Movement total latency (152 s) reflects packet transmission time (~2.5 min across five target hosts); detection latency was ~0 s once the cardinality threshold was breached. An operationally significant bug was discovered during TCP flag testing: the original rule contained a prefix query targeting an Elasticsearch ip-type field, causing silent shard failures that excluded the most recent daily index from search. Correcting the query to use term with CIDR notation resolved the issue and underscores that rule validation against actual field mappings is a required deployment step.

### 3.3. Multi - Channel Notification Results

All three notification channels email, Telegram, and Slack operated successfully throughout the evaluation period (Fig. 4). Measured delivery latencies: email 15-45 s (SMTP relay dependent), Telegram 3-8 s, Slack 2-5 s. These latencies satisfy real-time incident notification requirements for the threat categories evaluated.




a) b)

c) d)

*Figure 4. ElastAlert2 notification screenshots. (a) Telegram bot alert showing rule name, source/target IP, MITRE ATT&CK tag, and response steps. (b) Slack channel alert with severity color coding and structured event fields. (c) Email notification with full alert text and recipient. (d) Kibana Security Alerts dashboard showing triggered rules list with severity, risk score, and timestamp columns.*

### 3.4. IPS-Layer Detection: Suricata Integration Results

Table 6 presents the results of six controlled attack scenarios executed from the Kali Linux host (192.168.19.130) against the Metasploitable 2 target (192.168.11.135) through the OPNsense NAT boundary, with Suricata IPS operating in Divert mode unless otherwise noted.

For each scenario, the table records whether a Suricata alert was generated (and if so, the matched signature), whether the connection was confirmed in the OPNsense filterlog (opnsense-syslog-* index), and the ultimate exploit outcome. The results demonstrate a systematic detection gap at the IPS signature layer across post-exploitation and credential-based attack vectors, while confirming consistent network-layer visibility through the firewall logging path in all six scenarios. Figure 5 shows Kibana Security Alerts dashboard (alerts triggered after *nmap* scan).

*Table 6. Suricata IPS Detection Results Simulated Attack Scenarios Against Metasploitable 2*

| # | Attack Scenario | CVE / Technique | Trigger Tool | Suricata Alert | Suricata Signature | Filterlog Confirmed | Exploit Outcome | IPS Result |
|---|---|---|---|---|---|---|---|---|
| **1** | Web Recon / Path Traversal (port 80) App Reconnaissance (port 80) | CVE-2020-3452 (pattern) | Nikto | 29 | ET EXPLOIT Cisco ASA/Firepower Path Traversal M1/M3 (Severity 1) | Yes (pass) | Recon completed | True Positive |
| **2** | Tomcat Manager Brute Force (port 8180) | Default credentials (tomcat:tomcat) | Hydra | 0 | None | Yes - 43 hits (rdr + pass) | Valid credentials found | False Negative |
| **3** | UnrealIRCd 3.2.8.1 Backdoor RCE (port 6667) | CVE-2010-2075 | Metasploit | 0 | None | Not separately confirmed | Interactive shell obtained (uid=0) | False Negative |
| 4 | vsftpd 2.3.4 Backdoor IPS active (ports 21, 6200) | CVE-2011-2523 | Metasploit | 0 (IPS active) | None | Yes - block action, ElastAlert2 triggered | Session blocked - no shell | Incidental Prevention (Divert mode) |
| 5 | vsftpd 2.3.4 Backdoor IPS disabled (ports 21, 6200) | CVE-2011-2523 | Metasploit | N/A (IPS offline) | N/A | Yes - 32 hits (rdr + pass, ports 21 and 6200) | Interactive shell obtained (uid=0, root) | Undetected - IPS blind spot |
| 6 | ingreslock Backdoor Root Shell (port 1524) | No CVE (persistent backdoor) | nc | 0 | None | Yes - 2 hits (rdr + pass) | Interactive root shell obtained | False Negative |

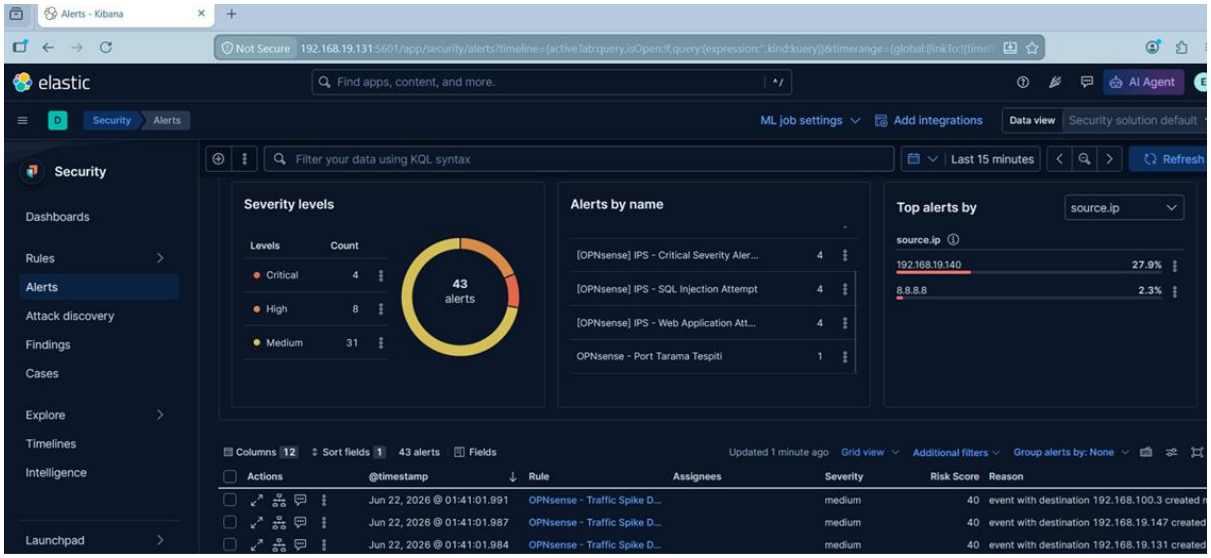



*Figure 5. Kibana Security Alerts dashboard (alerts triggered after nmap scan)*

Web application reconnaissance (Nikto, port 80) produced the only IPS true positive: 29 Severity-1 alerts matching a Cisco ASA path traversal signature (CVE-2020-3452). The match was behavioral Nikto's URL patterns triggered the signature despite targeting an Apache server rather than Cisco hardware correctly identifying the path traversal category while misattributing the CVE. This illustrates a known property of broad behavioral signatures.

All four exploitation scenarios that obtained interactive root shells (rows 2, 3, 5, 6) produced zero Suricata alerts, despite full filterlog visibility. The pattern is consistent: ET Open rulesets are optimized for web-application and malware traffic, and systematically underrepresent authenticated brute-force flows, protocol-specific legacy backdoors, and bare TCP listeners.

The vsftpd scenario (rows 4-5) produced the most operationally instructive finding. With Suricata active, the Divert-mode traffic interception blocked the bind shell callback on port 6200 without a matching signature an architectural side-effect confirmed by an ElastAlert2 "Firewall Block Action" alert (Source: 192.168.19.130 →192.168.11.135:6200, PF Rule ID 85). Disabling Suricata allowed the exploit to complete: filterlog recorded 32 entries (rdr + pass, ports 21 and 6200) confirming end-to-end delivery, and an interactive root shell was obtained (uid=0, hostname:metasploitable, kernel: Linux 2.6.24-16-server). This demonstrates that inline IPS traffic diversion constitutes a security control independent of signature coverage.

### 3.5. Cost Analysis

The proposed architecture presents a significant cost advantage compared to commercial SIEM solutions available in the market. Table 7 compares the estimated total cost of ownership (TCO) of the open - source architecture with representative commercial alternatives.

*Table 7. Estimated annual cost comparison open - source SIEM architecture vs. commercial solutions*

| Solution | License Cost (USD/year) | Hardware / Infrastructure | Total TCO (USD/year) |
|---|---|---|---|
| **This Work (ELK + OPNsense)** | 0 (Open Source) | ~500 - 1,000 | ~500 - 1,000 |
| **Splunk Enterprise (SMB tier)** | 15,000 - 50,000 | ~2,000 - 5,000 | 17,000 - 55,000 |
| **IBM QRadar (Base)** | 20,000 - 80,000 | ~3,000 - 8,000 | 23,000 - 88,000 |
| **Microsoft Sentinel (1 GB/day)** | ~3,000 - 6,000 | Azure cloud costs | ~5,000 - 12,000 |

*Note: Commercial pricing is estimated from publicly available sources; actual quotes vary significantly by institution size, negotiated contracts, and configuration.*

## 4. Discussion

### 4.1. Strengths of the Architecture

All core components are available without licensing fees, enabling security infrastructure investment where IT budgets are constrained. Elasticsearch natively supports multi-node cluster deployment, allowing straightforward scaling as data volume grows. The Logstash pipeline accommodates new log sources through additional .conf files, preserving modularity. The two-stage Bash installation scripts enable complete environment reconstruction on any Ubuntu 24.04 LTS server in minutes, supporting both disaster recovery and multi-site deployment.

### 4.2. Limitations and Potential Improvements

The single-node Elasticsearch deployment lacks high availability; production environments should use a minimum three-node cluster. The log source scope is limited to a single firewall; real enterprise environments additionally require Windows Event Logs (Winlogbeat), Linux Auditd (Auditbeat), and cloud service telemetry. Integration with an open-source SOAR platform (Shuffle or TheHive) would extend the architecture from detection to automated incident response.

Machine learning-based anomaly detection through Elasticsearch X-Pack ML or standalone Python libraries would address the signature coverage gap demonstrated in Section 3.4, particularly for zero-day and insider threat scenarios.

### 4.3. Security and Ethical Considerations

All experiments were conducted in an isolated VMware-based virtual infrastructure. Network traffic was generated and captured exclusively within this sandboxed environment, ensuring that no production systems or external networks were affected. The OPNsense filterlog structure was validated to contain only network-layer metadata (IP addresses, ports, protocol identifiers, and firewall actions), without any personally identifiable information or application-layer user content. Access to Elasticsearch and Kibana was restricted to the controlled lab environment, with authentication and encryption mechanisms enabled to safeguard against unauthorized access.

## Conclusion

This study demonstrated that enterprise-grade SOC capability is achievable for small and medium-sized organizations using exclusively open-source components. The proposed architecture OPNsense with Suricata IPS, ELK Stack, and ElastAlert2 was deployed and evaluated through rule-based detection testing and controlled penetration testing.

Performance measurements confirmed ~1.1 s log delivery latency, ~8,500 events/min peak throughput, and 3–50 s alert latencies. All eight ElastAlert2 rules triggered correctly; the traffic spike rule distinguished benign ~1.6× increases (true negative) from genuine ~4.4× spikes (true positive).

The penetration testing evaluation revealed a systematic IPS signature gap: 29 Suricata alerts were generated from web reconnaissance, while four scenarios yielding interactive root shells produced zero IPS alerts despite full filterlog visibility.

The vsftpd scenario demonstrated that Suricata's Divert mode provides incidental protection beyond its signature scope, and that failed exploitation attempts generate actionable SIEM intelligence through firewall-layer logging. These findings validate the defense-in-depth design: firewall logging, IPS signatures, and behavioral ElastAlert2 rules provide complementary not redundant detection coverage.

The architecture achieves a 96–98% TCO reduction relative to entry-level commercial SIEM platforms while meeting real-time monitoring requirements on modest hardware. Future work should address single-node resilience, broader log source integration, SOAR automation, and machine learning-based anomaly detection to extend coverage beyond the signature boundaries documented in this evaluation.

**References:**


[ 1.] Wazuh SIEM for Cyber Security and Threat Mitigation in Apparel Industries. (2024). International Journal of Engineering Materials and Manufacture, 9(4), 136 - 144. https://doi.org/10.26776/ijemm.09.04.2024.02

[ 2.] López Velásquez, J.M., Martínez Monterrubio, S.M., Sánchez Crespo, L.E. et al. Systematic review of SIEM technology: SIEM - SC birth. Int. J. Inf. Secur. 22, 691 - 711 (2023). https://doi.org/10.1007/s10207 - 022 - 00657 - 9

[ 3.] H. Mokalled, R. Catelli, V. Casola, D. Debertol, E. Meda and R. Zunino, "The Applicability of a SIEM Solution: Requirements and Evaluation," 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Napoli, Italy, 2019, pp. 132 - 137, doi: 10.1109/WETICE.2019.00036.

[ 4.] Vardalachakis, M., Vasilakis, M., & Tampouratzis, M. (2025). A Comprehensive Analysis of Features, Benefits, Challenges, and Best Practices of Security Information and Event Management (SIEM) Solutions. Computer Sciences & Mathematics Forum, 12(1), 18. https://doi.org/10.3390/cmsf2025012018

[ 5.] Gnatyuk, S., Berdibayev, R., Sydorenko, V., Polozhentsev, A., & Ryabyy, M. (2022). Enterprise Service Bus Construction in SOA Architecture for SIEM Implementation in Critical Information Infrastructure. In CPITS (pp. 11 - 20).

[ 6.] Hase, L. (2024). The Path to Choosing a SIEM System - A Systematic Literature Review. Semin. IT - Management Digit. Age, 1 - 12.

[ 7.] Farrel, F. I. F., Mardianto, I., Kom, M., & Qamar, M. I. A. S. (2024). Implementation of security information & event management (siem) wazuh with active response and telegram notification for mitigating brute force attacks on the gt - i2ti usakti information system. The GT - I2TI USAKTI Information System. Intelmatics, 1 - 7.

[ 8.] Vosshenrich, J., Breit, H. C., Bach, M., & Merkle, E. M. (2022). Economic aspects of low - field magnetic resonance imaging: acquisition, installation, and maintenance costs of 0.55 T systems. Der Radiologe, 62(5), 400.

[ 9.] Andronache, M. M., Vulpe, A., & Burileanu, C. (2025, January). A Comparative Study of Intrusion Events in Different SIEM Systems. In 2025 IEEE 23rd World Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 000065 - 000070). IEEE.

[ 10.] Amami, R., Charfeddine, M., & Masmoudi, S. (2024, July). Exploration of Open Source SIEM Tools and Deployment of an Appropriate Wazuh - Based Solution for Strengthening Cyberdefense. In 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 1 - 7). IEEE.

[ 11.] Usman Bilal, M., Waseem Iqbal, M., & Irfan, D. (2023, May). Analysis of ICT Infrastructure wrt Performance and Security Perspective Using SIEM Capabilities. In International Conference on Computing & Emerging Technologies (pp. 415 - 431). Cham: Springer Nature Switzerland.